PATENT
Docket No. **SF/0014.01**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Sethuraman Suresh et al.

Serial No.: 08/923,612

Filed: September 4, 1997

For: System and Methods for
Synchronizing Information Among
Disparate Datasets

Examiner: Channavajjala, S.

Art Unit: 2776

APPEAL BRIEF

Box AF, Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

BRIEF ON BEHALF OF SETHURAMAN SURESH ET AL.:

This is an appeal from the Final Rejection mailed October 14, 1999, in which currently-pending claims 1-25 and 27-30 stand finally rejected. Appellants filed a Notice of Appeal on March 20, 2000 (as indicated by return of a confirmation postcards stamped "OIPE MAR 20 2000"). This brief is submitted in triplicate in support of Appellants' appeal.

TABLE OF CONTENTS

1. REAL PARTY IN INTEREST:

The real party in interest is assignee Starfish Software, Inc., a California corporation, located at 1700 Green Hills Road, Scotts Valley, CA 95066.

2. RELATED APPEALS AND INTERFERENCES:

There are no appeals or interferences known to Appellants, the Appellants' legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

3. STATUS OF CLAIMS:

Claims 1-25 and 27-30 are pending in the subject application and are the subject of this appeal. An appendix setting forth the claims involved in the appeal is included as the last section of this brief.

4. STATUS OF AMENDMENTS:

One Amendment has been filed in this case. Appellants mailed an Amendment on August 4, 1999, in response to a non-final Office Action dated February 18, 1999. In the Amendment, the pending claims were amended in a manner which Appellants believe clearly distinguished the claimed invention over the art of record, for overcoming the art rejections. In response to the Examiner's Final Rejection dated October 14, 1999, Appellants filed a Notice of Appeal. Appellants have chosen to forego filing an Amendment After Final which might further limit Appellants' claims, as it is believed that further amendments to the claims are not warranted in view of the art. Accordingly, no Amendments have been entered in this case after the date of the Final Rejection.

As to matters of formality first raised in the Final Rejection, the Examiner has objected to the Specification's paper source code appendix. As Appellants intend to cancel the paper source code appendix in favor of a microfiche copy of that appendix upon indication of allowable subject matter, it is requested that the objection be held in abeyance. Additionally, the Examiner has noted a typographical error in the second-submitted Information Disclosure Statement (Paper #6, mailed August 23, 1999), which was a resubmission for the purpose of paying the fee under 37 CFR Section 1.17(p) since the first-submitted Information Disclosure

Statement (Paper #5) crossed the first Office Action in the mail. The Examiner has correctly discerned the relevant patent (U.S. Patent Number 5,684,990) from the first-submitted Information Disclosure Statement. Appellants confirm that the listing of U.S. Patent Number 5,694,990 was in error, as the Examiner has already surmised from the first-submitted Information Disclosure Statement (as well as the fact that a copy of U.S. Patent Number 5,684,990 was the Boothby patent provided in both papers).

5. SUMMARY OF INVENTION:

Appellants have invented a synchronization system with methods for synchronizing information among disparate datasets. In basic operation, the core engine of the synchronization system issues requests to accessors components which, in turn, instruct the relevant device (i.e., data-storing/processing device) to perform an appropriate data item or record operation, such as inserting a new record, or updating or deleting an existing record. The core engine asks the accessors (operating in conjunction with a conduit component) to provide those records that have changed since the last synchronization. Each such record is provided in the form of an extract record. Each extract record is processed according to the outbound synchronization logic and, if it has not been removed, is then mapped to a record map which provides a global identifier or ID and a timestamp.

Global IDs, which are employed at the level of a Record Map, are global to the entire synchronization system. Even if a source dataset already provides unique IDs, those unique IDs are generally unique to the device or unique to particular records on that device. Here, the synchronization system provides a unique global identifier (e.g., 32-bit or 64-bit value) for each data item at the level of the Record Map. Each global identifier can be based on an existing identifier, such as an existing record ID (RID) value, or can be synthesized de novo at runtime (e.g., based on system time/date information), particularly in those instances where the underlying dataset does not provide any identifiers for its data items. Regardless of how a global ID is provided for a given data item, that global ID is employed throughout the synchronization process for supporting synchronization across multiple devices (e.g., from palmtop to desktop to server) without creating duplicates. In conjunction with the system storing global IDs for identifying particular data items, the system also tracks when each data item or record is last modified (or created) by storing a last-modified date stamp. The actual change which occurred is logged in the Record Map. Internally, therefore, the identification of each record is tracked together with information about when each record was last changed, so that proper synchronization can be carried out across multiple devices.

The basic flow or propagation of a record change from one dataset to another is as follows. At the outset, the changed record is extracted and looked up in the record map. The addition of the new record is noted in a Transaction Table, so the system can determine whether the record has already been dealt with. A corresponding export record (i.e., record data along with a globally unique ID) is generated for inclusion in the synchronization set or SyncSet being created. The SyncSet contains all of the record actions that need to be transmitted to the other device, including insertions, updates, or deletions. The SyncSet can, at this point, be sent via a variety of transport mechanisms, including e-mail, FTP (file transport protocol), and two-way pagers. The receiving device processes the SyncSet by proceeding to map it, during inbound synchronization. Here, the corresponding import record is mapped into a record form for the target dataset. Once the appropriate records get inserted, updated, or deleted, the inbound Transaction Table is updated. In this fashion, the present invention provides a synchronization system and methodology that allows each device to remain synchronized with all other devices in a convenient, transparent manner.

6. ISSUES:

The issues presented on appeal are: (1) whether claims 1-13 should be rejected under 35 U.S.C. Section 103(a) as being unpatentable over Kucala (U.S. Patent No. 5,727,202) in view of Olds et al. (U.S. Patent No. 5,832,487); (2) whether claims 14-17 should be rejected under 35 U.S.C. Section 103(a) as being unpatentable over Kucala (U.S. Patent No. 5,727,202) in view of Olds et al. (U.S. Patent No. 5,832,487), and further in view of Buchanan (U.S. Patent No. 5,758,355); and (3) whether claim 21-25 and 27-30 should be rejected under 35 U.S.C. Section 103(a) as being unpatentable over Meyering (U.S. Patent No. 5,729,735) in view of Olds et al. (U.S. Patent No. 5,832,487).

7. GROUPING OF CLAIMS:

For purposes of this appeal, Appellants believe that the following groups of claims are separately patentable under Section 103. Thus, the claims do not stand or fall together with respect to Section 103 but are instead grouped as follows:

Group I:	claims 1-13
Group II:	claims 14-17
Group III:	claims 21-25 and 27-30

8. ARGUMENT:

A. First rejection under 35 U.S.C. Section 103(a): Kucala in view of Olds et al.

1. General

Under Section 103(a), a patent may not be obtained if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which the subject matter pertains. To establish a prima facie case of obviousness under this section, the Examiner must establish: (1) that there is some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings, (2) that there is a reasonable expectation of success, and (3) that the prior art reference (or references when combined) must teach or suggest all the claim limitations. (See e.g., MPEP 2142). The references cited by the Examiner fail to meet these conditions.

2. Group I claims

Claims 1-13 stand rejected under 35 U.S.C. Section 103(a) as being unpatentable over Kucala in view of Olds et al. (hereinafter, "Olds").

Kucala describes a fairly simplistic synchronization approach that uses backup files to reflect the status (e.g., new, updated, or deleted) of records on a palmtop device (e.g., Palm handheld computing device), as well as to reflect the status of records on a desktop device (e.g., PC computer). The results of both compares, which are stored in a temporary data structure (e.g., a "reconcile file"), are copied over the selected files on the palmtop, the PC, and the backup file. Thus for the claims in this group, the Examiner refers to Kucala as providing a basic teaching of a synchronization system (i.e., a system for synchronizing source and target data sets).

The Examiner acknowledges, though, that Kucala does not teach Appellants' limitation "wherein each information record of the source dataset is assigned a globally unique identifier that is independent of either of the devices, for identifying said each information record at both the source dataset and the target dataset, said globally unique identifier being maintained in a device-independent record map that allows the globally unique identifier to be traced back to a specific information record regardless of which device the specific information record resides." The Examiner, nevertheless, turns to Olds as providing this missing feature. Here, the Examiners states:

> Olds teaches a system which including 'source dataset' [col 3, line 20-24], examiner interpreting source dataset to be equivalent to Olds's file servers, element 16, 'source dataset is assigned a globally unique identifier that is independent of either of the devices for identitying said each information record at both source dataset and the target dataset'[col 3, line 25-30, col , col 6, line 64-67], 'at both source dataset and the target dataset' [col 3, line 42-45], examiner interpreting target dataset to be equivalent to Olds's Clients, element 20, 'globally unique identifier being maintained' [col 8, line 17-19], globally unique identifier to be traced back to a specific information record regardless of which device the specific information record resides'[col 8, line 24-34] 'globally unique identifiers' [col 3, line 54-57], deleting from the target dataset any information records which have been previously transmitted to the target dataset but no longer exist at the source dataset'[col 10, line 7-14, line 19-28], 'using said globally unique identifiers, updating the target dataset so that said target dataset includes those information records determined to have added to or modified at the source dataset since the source dataset was last synchronized with the target dataset'[col 10, line 15-28].

Thus, according to the Examiner, Olds teaches Appellants' approach of assigning each information record of the source dataset a device-independent globally unique identifier that allows Appellants' synchronization system to track each record regardless of what device or environment each record ultimately resides. A detailed review of Olds, however, does not bear out the Examiner's position.

Olds describes an approach "for managing replicated objects in a partitioned hierarchical database" (particularly, NetWare Directory Services database). The approach "combines partition-wide object identifiers in order according to ancestry to form a database-wide object identifiers that identifies a target object relative to all other objects in the database. Each partition-wide object identifier includes a replica identifier and at least one integer value." The identifiers may be used "to access objects after a single object or a subtree of objects has been renamed or moved." (See, e.g., Olds' Abstract, at lines 1-11.)

As described by Olds, his approach is applicable to hierarchical network databases. These are tree-based structures employing nodes, branches, and levels for describing a network and its resources. For instance, the top node of the tree represents the entire network. The "leaves" of the tree represents network resources, including servers, printers, users, user groups, and so on. (See, e.g., Olds at column 1, lines 20-26, and lines 43-50.) As described by Olds at column 1, lines 57-60, "The hierarchical network database is stored on at least one server computer on the network. However, to avoid a single point of failure, a database copy or replica can be stored on one or more servers on the network." Since replicas of large network databases are expensive to store and keep updated, the databases often divided into smaller "partitions," each of which corresponds to a subtree of the database tree. Each copy of a partition is called a "replica." Typically, each node and leave of a tree is given a textual identifier, thereby indicating to a network administrator what portion of the network corresponds to that node. To name a resource at one of the leaves, one may concatenate the identifiers associated with the nodes along the path leading from the top node of the tree to the leaf. This results in an object identifier known as a "distinguished name," which is unique across that particular network (e.g., a particular company's NetWare network). (See, e.g., Olds at column 2, lines 4-15.)

Since object identifiers may change, Olds seeks to provide a more permanent way of naming objects in a network database, one which will survive the renaming of node and leaf identifiers. (See, e.g., Olds at column 2, lines 31-33.) The approach he adopts is to provide an object identifier for an object by starting with the relative identifier of an object and adding the relative identifier of each ancestor object until the relative identifier of the root of the object hierarchy is added:

> When an object is created within a replica it is assigned a unique identifier relative to its sibling objects, such that no two siblings in any replica of the partition have the same relative identifier. The relative identifier contains its replica number and is therefore guaranteed to be unique among all of the sibling objects created in all replicas of that partition. The invention provides a complete object identifier for an object by starting with the relative identifier of an object and adding the relative identifier of each ancestor object until the relative identifier of the root of the object hierarchy is added. Such a sequence of relative object identifiers can be used as a global identifier of an object and is referred as a 'tuned-name' or an 'absolute name.' (Olds at column 3, lines 25-38)

With an understanding of Olds, one may now contrast that approach with that provided by Appellants' invention.

Central to the Examiner's rejection is the premise that Olds' scheme ("tuned names") meet Appellants' device-independent, globally-unique identifiers maintained in a device-independent record map. A careful review of the foregoing elements of Olds' teachings does not bear out the Examiner's contention. From his description, Olds clearly describes that each copy of a subtree or "replica" resides within a single network hierarchical database, such as a NetWare Directory Services (NDS) database. As the single network hierarchical database is partitioned among multiple server computers connected to a network, each "replica" preserves its data structure for that system (e.g., preserves its data structure so that it can be used within the NDS database). For instance, if a given replica is, for purposes of distributed computing, replicated from one server computer to another, the underlying data structure of that replica remains unchanged (e.g., it remains an NDS subtree data structure).

This is quite different from the data records undergoing synchronization in Appellants' system, where a given data record -- in an original (native) format or data structure -- is transferred from an original device (e.g., handheld Palm device) to a totally-different target device (e.g., desktop computer) -- one that does not support that original format. Consider the problem described by Appellants' Background section.

> With each passing day, there is ever increasing interest in providing synchronization solutions for connected information appliances. Here, the general environment includes "appliances" in the form of electronic devices such as cellular phones, pagers, hand-held devices (e.g., Palm Pilot™ and Windows™ CE devices), as well as desktop computers and the emerging "NC" device (i.e., a "network computer" running, for example, a Java virtual machine or a browser).

> A problem facing such an environment today is that these devices do not communicate well with desktop computers, let alone with each other. In particular, **a problem exists as to how one integrates __disparate information__** -- such as calendaring, scheduling, and contact information -- **among __disparate devices__**. Consider, for instance, a user who has his or her appointments on a desktop PC but also has a battery-powered, hand-held device for use in the field. What the user really wants is for the information of each device to remain synchronized with all other devices in a convenient, transparent manner. Still further, the desktop PC is typically connected to a server computer, which stores information for the user. The user would of course like the information on the server computer to participate in the synchronization, so that the server also remains synchronized.
> (Appellants' Specification, page 1, line 21 - page 2, line 7; emphasis added)

Olds, in stark contrast, describes an approach involving replication of objects (e.g., subtrees) that preserves the underlying native format, as the objects are simply been replicated among uniformly similar machines (namely, server computers that support a single NetWare Directory Services (NDS) database).

The claim limitations present in Appellants' claims of this group bring this distinction to the forefront. Appellants' claim 1 requires that "each information record of the source dataset is assigned a globally unique identifier that is independent of either of the devices." Still further, the claim requires that the globally unique identifier be "maintained in a device-independent record map that allows the globally unique identifier to be traced back to a specific information record regardless of which device the specific information record resides." In this manner, the record map provides a means of identifying records to the synchronization system's logic, independent of how a record is identified within its own data set. This approach allows a given information record to be transferred from one device, say, a cellular phone, to a completely different device, say, a desktop computer, regardless of the fact that the receiving device does not support the original format for the information record. On the other hand, if one were simply transferring information records among like devices, such as like server computers as done by Olds, there would be no need to fashion an identifier that was independent of those devices; instead, one can simply rely on whenever __system-dependent__ unique identifier was already provided by that system.

Furthermore, Olds explicitly considers use of a timestamp identifier and a GUID (globally-unique ID) identifier. However, he dismisses both. (See, e.g., Olds at column 2, lines

39-44, and lines 45-55, respectively.) For a timestamp, for instance, Olds states that a "timestamp identifier does not help locate the object in the hierarchical database" (at column 2, lines 43-44). Similarly, for a globally-unique ID, Olds states that "GUIDs contain no particular information to help locate an object in the database." Olds' approach may prove to be a useful one given that his data objects of interest (e.g., NDS subtrees) all remain within a homogeneous environment (e.g., server computers supporting partitions of an NDS database), and therefore one may use incorporate system-dependent information into an identifier to help locate an object within that environment. Olds' approach is, however, specifically dependent on the original (native) format of the information (namely, the hierarchical organization employed in a NetWare Directory Services (NDS) database), in contrast to Appellants' device-independent identifier approach.

Moreover, the GUIDs themselves that are employed in Appellants' system may in fact be rather conventional identifiers, such as a 32-bit or 64-bit value quantity (e.g., integer). It is not so much the identifiers themselves that are of interest but how they are used during the synchronization process -- specifically, in a manner that makes each GUID <u>independent</u> of both the source and target devices. During operation of the present invention, the participating data sets (i.e., source and target data sets) typically will already have their own GUID values (e.g., assigned by their own computing device). In the claimed invention, the GUIDs, whether the aforementioned device-provided GUID values or newly synthesized values, are employed internally by the synchronization system -- at the level of the synchronization system's device-independent record map -- in a manner that allows each data item to be uniquely identify across <u>multiple disparate devices</u>, regardless of whether such devices may or may not be connected to a network and regardless of whether such devices have identification schemes, data formats, or the like that are compatible with one another. By tracking the identification of each record in a manner which is independent of what device or system the record resides and doing so at the level of the synchronization's system record map, Appellants' invention allows proper synchronization to be carried out across multiple <u>dissimilar</u> devices in a convenient, transparent manner.

The Examiner has attempted to recreate Appellants' claimed methodology by grafting onto Kucala the capabilities of a system having a system-dependent approach to naming objects (i.e., Olds). Nothing in Olds teaches or suggests that a system-dependent naming approach methodology could be extended to support device-independent synchronization capability. Olds, for its part, teaches use of a system-dependent naming scheme that expressly eschews use of GUIDs -- an approach that is contrary to Appellants' approach of using device-independent GUIDs. Moreover, the Examiner's proposed combination of Kucala with Olds still fails to re-create Appellants' invention as the references themselves fail to provide any teaching that would indicate how Kucala synchronization capability would be revamped for use with device-independent GUIDs stored in a record map (also, device-independent), so as to re-create Appellants' claimed invention. It is respectfully submitted that this chiasm can only be bridged by relying on the hindsight benefit of Appellants' specification to reconfigure Kucala's device to operate in a manner which it itself is not intended to support (and do so in a manner which is not even suggested by Olds).

All told, it is respectfully submitted that the references fail to establish prima facie obviousness for the claims of Group I. Accordingly, it is respectively submitted that the Examiner's rejection of the claims under Section 103 should not be sustained.

B. Second rejection under 35 U.S.C. Section 103(a): Kucala in view of Olds, and further in view of Buchanan

1. General

As to claims 14-17, the Examiner relies on the reasoning applied above with respect to Kucala in view of Olds, but adds Buchanan (U.S. Patent No. 5,758,355) for the purpose of modifying the foregoing references to add a teaching of a filter limitation.

2. Group II claims

Group II includes claims 14-17. These claims include the aforementioned claim limitations (or equivalent thereof) set forth above for the claims of Group I and, thus, are believed to be allowable over the art, for at least the reasons cited above pertaining to Kucala and Olds as applied against the claims of Group I above (the discussion of which is hereby incorporated into this section by reference). The claims of this group are grouped separately from Group I as the Examiner has applied the additional reference of Buchanan to claims of this group. Even when combined with Buchanan, however, the references still fail to provide sufficient teaching or suggestion to allow one to create Appellants' claimed invention.

Additionally, the claims are believed to be allowable for the following additional reasons. In addition to including the limitation of a "filter," the claims specify how a specific filter is applied in regards to the synchronization process. For instance, claim 15 specifies that the filter "comprises an outbound filter applied to information records prior to creation of the synchronization set." Claim 16, on the other hand, specifies that the filter "comprises an inbound filter applied to information records after creation of the synchronization set." Even if the Examiner interprets the "filter" limitation to be equivalent to selection criteria in a database query (e.g., SQL predicate), the Buchanan reference still fails to teach or suggest specific application of such a "filter" in relation to creation of a synchronization set, as required by Appellants' claims. Accordingly, it is submitted that the claims distinguished over the cited references and that the Examiner's rejection under Section 103 should not be sustained.

C. Third rejection under 35 U.S.C. Section 103(a): Meyering in view of Olds

1. General

As to claims 21-25 and 27-30, the Examiner relies on the reasoning applied above with respect to Olds, but cites Meyering (U.S. Patent No. 5,729,735) as providing the basic teaching of a synchronization system.

Meyering provides a relatively simplistic synchronization approach using backup files to reflect contents of a remote file at a particular point in time (i.e., when it was created or last synchronized). Actual synchronization involves a simple comparison between the master file, the remote file, and the backup file, to determine which file (i.e., remote or master) has a more current version of the data. The Examiner acknowledges that Meyering does not teach the limitation of assigning a globally unique identifier. However, the Examiner contends that Olds provides this missing element, as for instance applied against the claims of Group I.

2. Group III claims

Group III, which consists of claims 21-25 and 27-30, includes the aforementioned claim limitations (or equivalent thereof) set forth above for the claims of Group I and, thus, are believed to be allowable over the art, for at least the reasons cited above pertaining to Olds as applied against the claims of Group I above (the discussion of which is hereby incorporated into this section by reference). The claims of this group are grouped separately from Group I as the

Examiner has applied the reference of Meyering (instead of Kucala) to claims of this group. Even when this combination is made, however, the references still fail to provide sufficient teaching or suggestion to allow one to create Appellants' claimed invention.

In particular, Olds lacks sufficient teaching or suggestion that would allow one to transform the relatively simplistic synchronization system of Meyering into one which employs device-independent GUIDs in a (likewise, device-independent) record map, for supporting synchronization among multiple, disparate devices. Accordingly, it is respectfully submitted that the Examiner has failed to established obviousness based on these references and, therefore, that the Examiner's rejection of the claims of Group III should not be sustained.

9. CONCLUSION:

The present invention greatly improves the ease and efficiency of the task of synchronizing information among a variety of disparate devices, by providing a device-independent approach employing device-independent GUIDs maintained in a device-independent record map. In this manner, the record map provides a means of identifying records to the synchronization system's logic, independent of how a record is identified within its own dataset. It is respectfully submitted that the present invention, as set forth in the pending claims, sets forth a patentable advance over the art.

In view of the above, it is respectfully submitted that the Examiner's rejections under 35 U.S.C. Section 103 should not be sustained. If needed, Appellants' undersigned attorney can be reached at (408) 395-8819. For the fee due for this Appeal Brief, please refer to the attached **Fee Transmittal Sheet**. This Brief is submitted in triplicate.

Respectfully submitted,

Date: August 18, 2000

John A. Smart; Reg. No. 34,929
Attorney of record

708 Blossom Hill Rd., #201
Los Gatos, CA 95032-3503
(408) 395-8819; (408) 490-2853 FAX

SF/0014.01

10. APPENDIX OF CLAIMS ON APPEAL:

1. In a system providing one dataset in communication with another dataset, a method for synchronizing datasets comprising:

receiving a request specifying synchronization of information records of a source dataset residing on a first device with information records of a target dataset residing on a second device;

determining a synchronization set by:

(i) determining which, if any, information records have been previously transmitted to the target dataset but no longer exists at the source dataset, and

(ii) determining which, if any, information records have been added to or modified at the source dataset since the source dataset was last synchronized with the target dataset,

wherein each information record of the source dataset is assigned a globally unique identifier that is independent of either of the devices, for identifying said each information record at both the source dataset and the target dataset, said globally unique identifier being maintained in a device-independent record map that allows the globally unique identifier to be traced back to a specific information record regardless of which device the specific information record resides; and

based on said synchronization set, synchronizing information records of the source dataset with information records of the target dataset by:

(i) using said globally unique identifiers, deleting from the target dataset any information records which have been previously transmitted to the target dataset but no longer exist at the source dataset, and

(ii) using said globally unique identifiers, updating the target dataset so that said target dataset includes those information records determined to have been added to or modified at the source dataset since the source dataset was last synchronized with the target dataset.

2. The method of claim 1, wherein each dataset comprises a database table having a plurality of data records.

3. The method of claim 1, wherein each dataset comprises an electronic address book listing contact information.

4. The method of claim 1, wherein each dataset comprises an electronic schedule listing scheduling information.

5. The method of claim 1, wherein said globally unique identifiers are created by the system regardless of whether the source dataset includes existing record identifiers.

6. The method of claim 5, wherein said globally unique identifiers are maintained in a record map stored apart from the source dataset.

7. The method of claim 1, wherein each said globally unique identifier for each record comprises a timestamp which is assigned to the record when thc rccord is initially processed by the system.

8. The method of claim 1, wherein each globally unique identifier is a 32-bit value.

9. The method of claim 1, further comprising:

synchronizing the information records of the target dataset with information records of the source dataset by designating the source dataset as the target dataset, designating the target dataset as the source dataset, and repeating said determining step and said synchronizing step.

10. The method of claim 1, wherein said synchronization set comprises a delete order specifying particular information records to delete at the target dataset.

11. The method of claim 10, wherein said delete order includes a list of globally unique identifiers for particular information records to delete at the target dataset.

12. The method of claim 1, wherein said synchronization set comprises an extraction record specifying particular information to add to or modify at the target dataset.

13. The method of claim 12, wherein said extraction record includes at least one globally unique identifier together with field information for the particular information to add to or modify at the target dataset.

14. The method of claim 1, further comprising:

excluding certain information records from participating in synchronization by applying a user-defined filter.

15. The method of claim 14, wherein said user-defined filter comprises an outbound filter applied to information records prior to creation of the synchronization set.

16. The method of claim 14, wherein said user-defined filter comprises an inbound filter applied to information records after creation of the synchronization set.

17. The method of claim 14, wherein said user-defined filter comprises a user-supplied filtering routine supplying filtering logic.

18. The method of claim 1, wherein said target dataset resides at a remote location relative to the source dataset.

19. The method of claim 18, further comprising:

after creating the synchronization set, transmitting said synchronization set to said remote location.

20. The method of claim 19, wherein the synchronization set is transmitted to the remote location using an electronic messaging communication protocol.

21. A synchronization systcm comprising:

means for connecting a first device having a first dataset to a second device having a second dataset;

means for determining information of said first and second datasets which requires synchronization, said means including:

(i) means for determining for each dataset information which has been previously received from the other dataset but which no longer exists at the othcr dataset, and

(ii) means for determining for each dataset information which has been added or modified at the other dataset since the other dataset was last synchronized with said each dataset; and

means, responsive to said determining means, for synchronizing said first and second datasets;

wherein said information of said first and second datasets comprises data records and wherein said means for determining include means for assigning to each data record a device-independent globally unique identifier created by the system for uniquely identifying each data record regardless of which dataset and device it appears.

22.. The system of claim 21, wherein at least one of said devices is a hand-held computing device.

23. The systcm of claim 21, wherein at least one of said devices is desktop computing device.

24. The system of claim 21, wherein said means for connecting includes a Transmission Control Protocol/ Internet Protocol (TCP/IP) connection.

25. The system of claim 21, wherein said means for synchronizing operates to provide bi-directional synchronization of the datasets.

26. --CANCELED--

27. The system of claim 21, further comprising:

filter means for selectively blocking synchronization of certain types of information.

28. The system of claim 27, wherein said filter means operates based on how old information is.

29. The system of claim 27, wherein said filter means operates based on particular information content.

30. The system of claim 21, further comprising:

electronic mail transport means for enabling synchronization of remote datasets.